RUST

Patrick Gilmore
Department of Computer Science
University of Arizona
Tucson, Arizona 85721
pmgilmore41@email.arizna.edu

Jack Lyons
Department of Computer Science
University of Arizona
Tucson, Arizona 85721
jacklyons@email.arizona.edu

March 24, 2020

ABSTRACT

1 Introduction

2 History

2.1 Why?

Most systems applications are written in C or C++ but as many masters of C know, C can have tons of Pitfalls. To solve this problem Rust was developed so that developers could write high Performance code that would normally need to be written in C or C++. Now with Rust developers need to worry much less about memory failure and code failing as a result of a segmentation fault.

2.2 Who?

The development of Rust was started by Graydon Hoare, a self proclaimed language engineer, in 2006 has a personal project. Down the line Mozilla took an interest in the project and put together a team to assist in the development of Rust. Mozilla continued to back Rust with the aim to rebuild their browser stack. To do so Mozilla aimed to replace the difficult C++ code with something that was easier to debug, write, and maintain efficiency.

2.3 Rust Today

Rust is still going strong to this day as indicated by stackoverflow.com, a popular site for developers to learn and ask questions about technology. In fact stackoverflow has voted Rust to by the most loved language 4 years in a row. Rust's drive to reduce the number of pains that developers face while writing system code in C++ while minimizing the consequences has been pulling in frustrated developers for years. So long as a developer does not need to make an application that goes way deep into the fundamentals of a system, Rust will permit easy convenient code.

3 Control Structures

4 Data Types

5 Subprograms

6 Summary

References

[1] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.

[2] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.

[3] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.